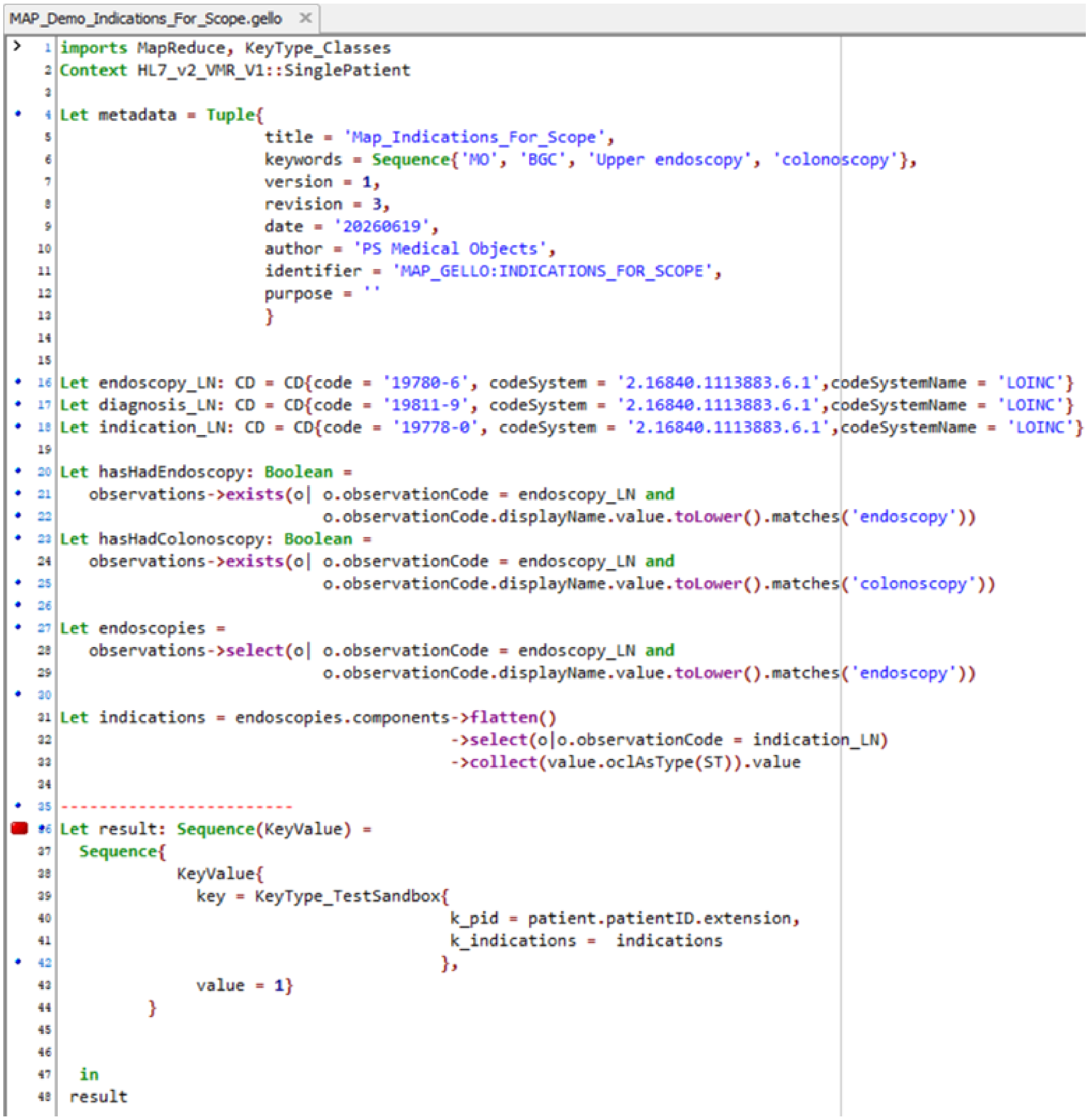
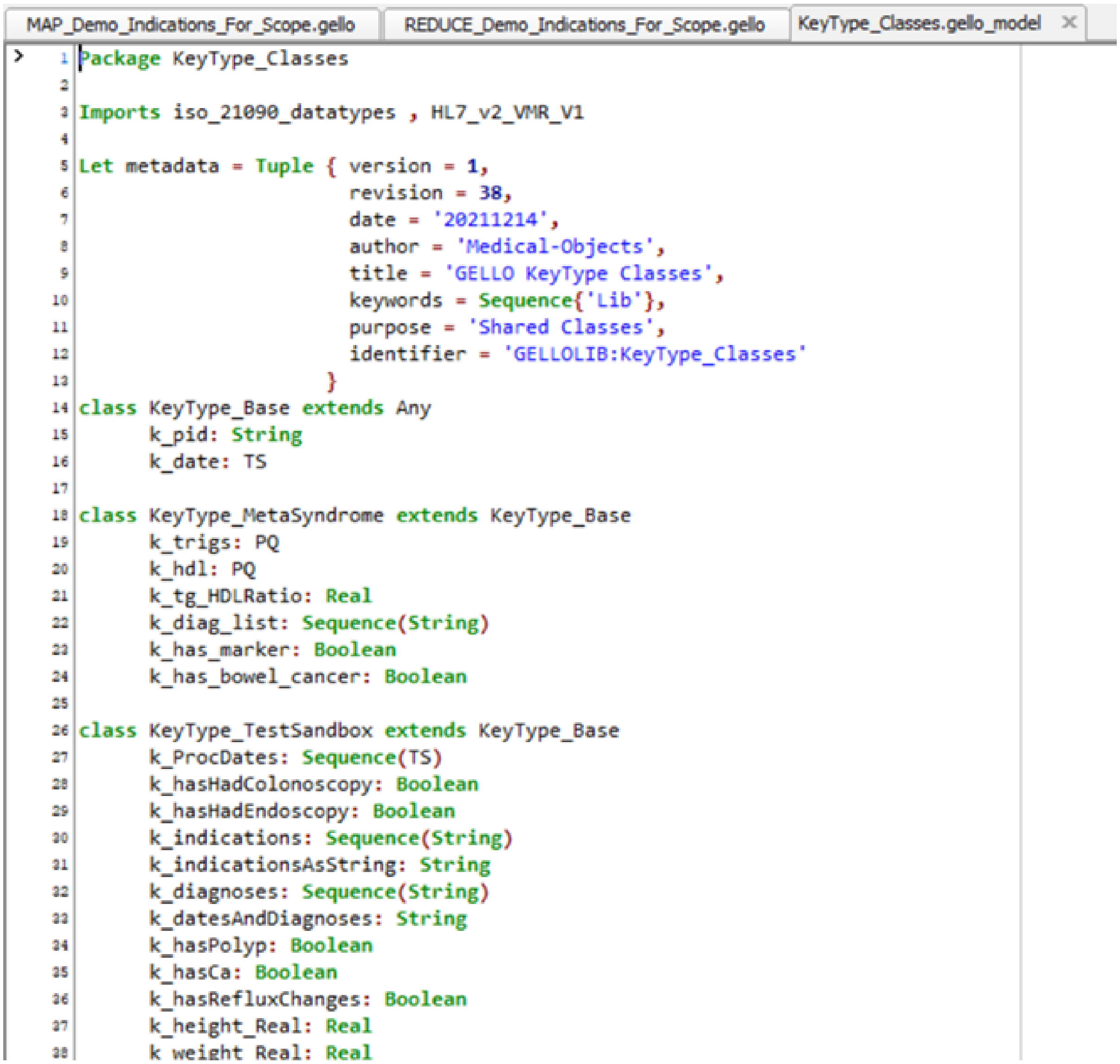
Population GELLO MapReduce Example

The Map gello looks similar to standard query with the result being Key Value pairs.



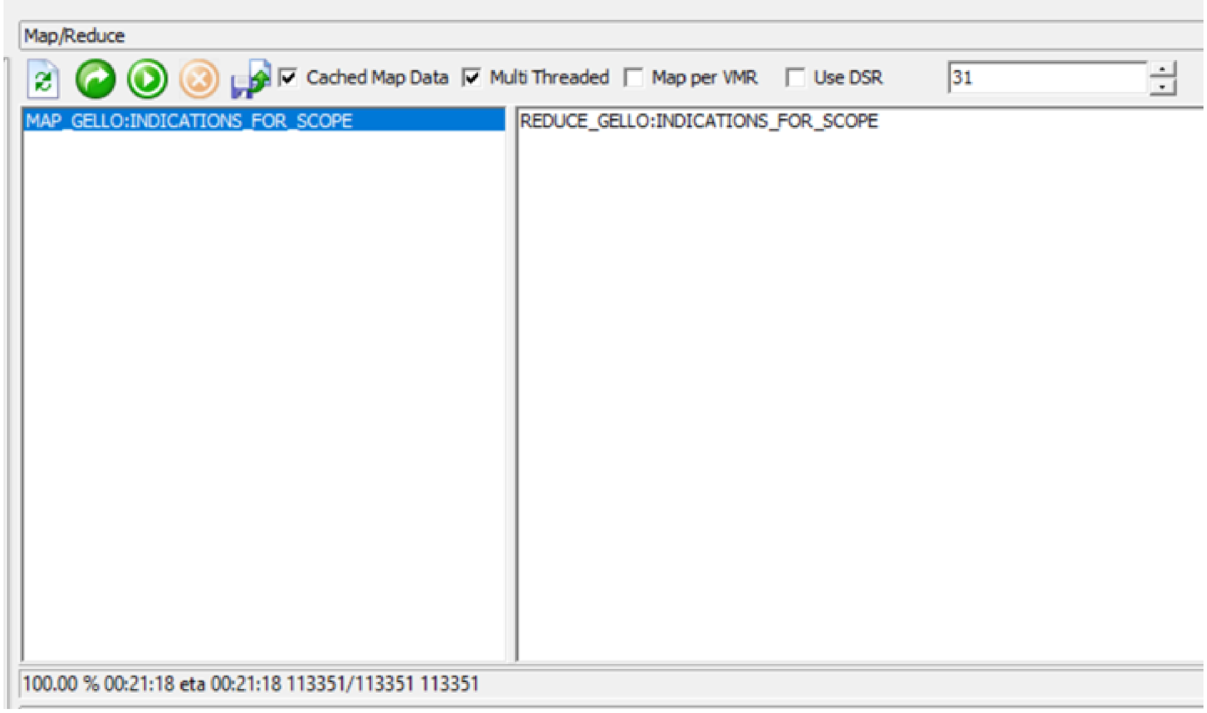
```
imports MapReduce, KeyType_Classes
Context HL7_v2_VMR_V1::SinglePatient

Let metadata = Tuple{
                     title = 'Map_Indications_For_Scope',
                     keywords = Sequence{'MO', 'BGC', 'Upper endoscopy', 'colonoscopy'},
                     version = 1,
                     revision = 3,
                     date = '20260619',
                     author = 'PS Medical Objects',
                     identifier = 'MAP_GELLO:INDICATIONS_FOR_SCOPE',
                     purpose = ''
                     }


Let endoscopy_LN: CD = CD{code = '19780-6', codeSystem = '2.16840.1113883.6.1',codeSystemName = 'LOINC'}
Let diagnosis_LN: CD = CD{code = '19811-9', codeSystem = '2.16840.1113883.6.1',codeSystemName = 'LOINC'}
Let indication_LN: CD = CD{code = '19778-0', codeSystem = '2.16840.1113883.6.1',codeSystemName = 'LOINC'}

Let hasHadEndoscopy: Boolean =
   observations->exists(o| o.observationCode = endoscopy_LN and
                           o.observationCode.displayName.value.toLower().matches('endoscopy'))
Let hasHadColonoscopy: Boolean =
   observations->exists(o| o.observationCode = endoscopy_LN and
                           o.observationCode.displayName.value.toLower().matches('colonoscopy'))

Let endoscopies =
   observations->select(o| o.observationCode = endoscopy_LN and
                           o.observationCode.displayName.value.toLower().matches('endoscopy'))

Let indications = endoscopies.components->flatten()
                                        ->select(o|o.observationCode = indication_LN)
                                        ->collect(value.oclAsType(ST)).value

------------------------
Let result: Sequence(KeyValue) =
  Sequence{
            KeyValue{
              key = KeyType_TestSandbox{
                                        k_pid = patient.patientID.extension,
                                        k_indications =  indications
                                       },
              value = 1}
         }


  in
 result
```

The keys are customisable.



```
Package KeyType_Classes

Imports iso_21090_datatypes , HL7_v2_VMR_V1

Let metadata = Tuple { version = 1,
                       revision = 38,
                       date = '20211214',
                       author = 'Medical-Objects',
                       title = 'GELLO KeyType Classes',
                       keywords = Sequence{'Lib'},
                       purpose = 'Shared Classes',
                       identifier = 'GELLOLIB:KeyType_Classes'
                     }
class KeyType_Base extends Any
      k_pid: String
      k_date: TS

class KeyType_MetaSyndrome extends KeyType_Base
      k_trigs: PQ
      k_hdl: PQ
      k_tg_HDLRatio: Real
      k_diag_list: Sequence(String)
      k_has_marker: Boolean
      k_has_bowel_cancer: Boolean

class KeyType_TestSandbox extends KeyType_Base
      k_ProcDates: Sequence(TS)
      k_hasHadColonoscopy: Boolean
      k_hasHadEndoscopy: Boolean
      k_indications: Sequence(String)
      k_indicationsAsString: String
      k_diagnoses: Sequence(String)
      k_datesAndDiagnoses: String
      k_hasPolyp: Boolean
      k_hasCa: Boolean
      k_hasRefluxChanges: Boolean
      k_height_Real: Real
      k_weight_Real: Real
```

The Maps are run on the whole population to produce a map cache. This example of the indications for endoscopy for a gastroenterology practice shows the map stage took about 20 minutes for 113000 nested and at times complex patient records as VMRs.



☑ Cached Map Data ☑ Multi Threaded ☐ Map per VMR ☐ Use DSR

31

MAP_GELLO:INDICATIONS_FOR_SCOPE

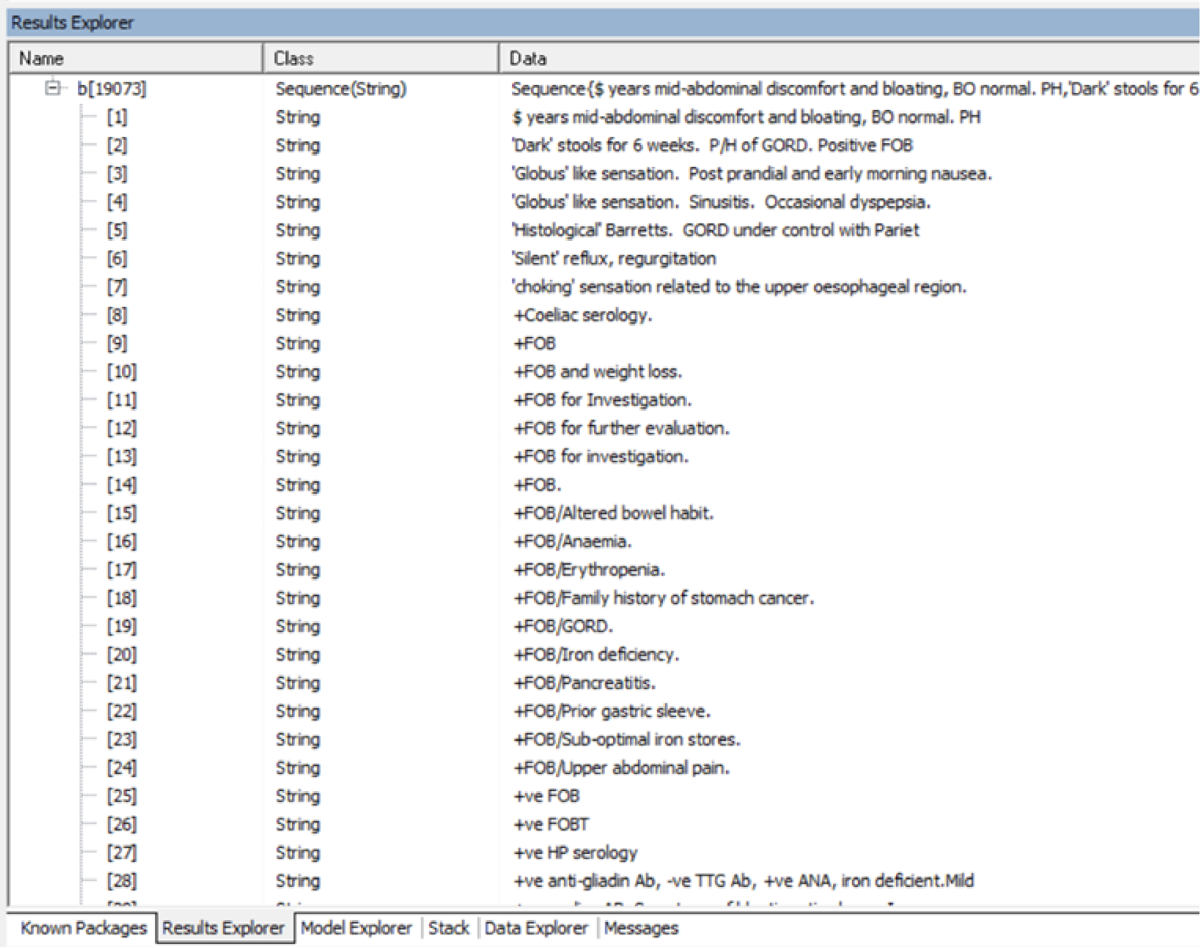
REDUCE_GELLO:INDICATIONS_FOR_SCOPE

```
100.00 % 00:21:18 eta 00:21:18 113351/113351 113351
```

Some Reduce gello is shown here; again its a fairly simple worked example. The loop is just to help the output strings be unique and somewhat cleaned.:

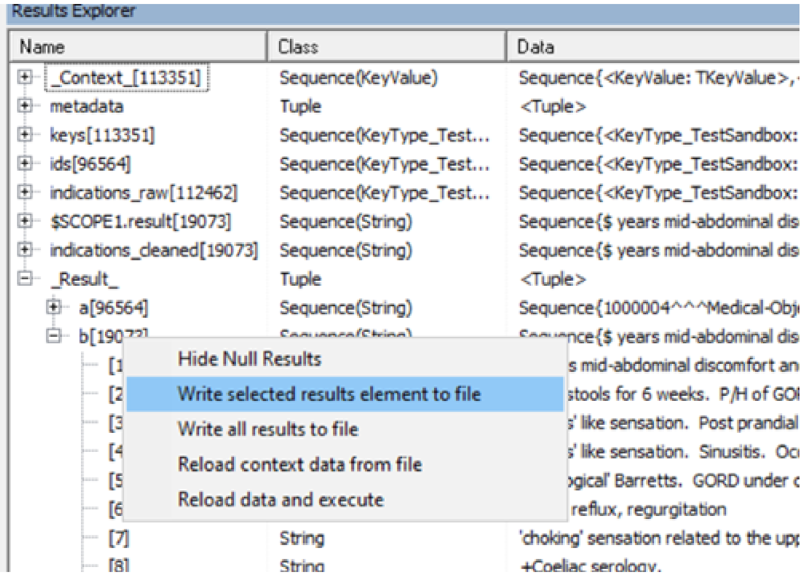
```
Imports MapReduce, HL7_v2_VMR_V1, KeyType_Classes
Package Local
 --functions
   UniqueStrings_Internal(i:Integer, accum: Sequence(String), strList:Sequence(String)): Sequence(String) =
    If i <= strList.size()
           then
            Let result = UniqueStrings_Internal(i+1,
                                                    If accum.excludes(strList[i]) and strList[i].matchPos('"') = 0
                                                    then accum->append(strList[i]) else accum endif ,
                                                    strList
                                                    )
          in
      result
    else
      accum
    endif

  UniqueStrings(strList: Sequence(String)): Sequence(String) =
      UniqueStrings_Internal(1, Sequence{}, strList)

EndPackage
Context Sequence(KeyValue)

Let metadata = Tuple{
                     title = 'indications For Scope',
                     keywords = Sequence{'MO', 'BGC', 'Upper endoscopy', 'colonoscopy'},
                     version = 1,
                     revision = 8,
                     date = '20260620',
                     author = 'PS Medical Objects',
                     identifier = 'REDUCE_GELLO:INDICATIONS_FOR_SCOPE',
                     purpose = '' ,
                     map_from = 'MAP_GELLO:INDICATIONS_FOR_SCOPE'
                     }

Let keys: Sequence(KeyType_TestSandbox) = Self->collect(key.oclAsType(KeyType_TestSandbox))
Let ids: Sequence(KeyType_TestSandbox) = keys->select(k_pid.oclIsDefined())
Let indications_raw: Sequence(KeyType_TestSandbox) = keys->select(k_indications.oclIsDefined() )
Let indications_cleaned = UniqueStrings(indications_raw.k_indications->flatten()->sort())
in

Tuple{
      a=ids.k_pid,
      b=indications_cleaned

}
```

The actual result for the combined MapReduce:



Name	Class	Data
b[19073]	Sequence(String)	Sequence{$ years mid-abdominal discomfort and bloating, BO normal. PH,'Dark' stools for 6
[1]	String	$ years mid-abdominal discomfort and bloating, BO normal. PH
[2]	String	'Dark' stools for 6 weeks. P/H of GORD. Positive FOB
[3]	String	'Globus' like sensation. Post prandial and early morning nausea.
[4]	String	'Globus' like sensation. Sinusitis. Occasional dyspepsia.
[5]	String	'Histological' Barretts. GORD under control with Pariet
[6]	String	'Silent' reflux, regurgitation
[7]	String	'choking' sensation related to the upper oesophageal region.
[8]	String	+Coeliac serology.
[9]	String	+FOB
[10]	String	+FOB and weight loss.
[11]	String	+FOB for Investigation.
[12]	String	+FOB for further evaluation.
[13]	String	+FOB for investigation.
[14]	String	+FOB.
[15]	String	+FOB/Altered bowel habit.
[16]	String	+FOB/Anaemia.
[17]	String	+FOB/Erythropenia.
[18]	String	+FOB/Family history of stomach cancer.
[19]	String	+FOB/GORD.
[20]	String	+FOB/Iron deficiency.
[21]	String	+FOB/Pancreatitis.
[22]	String	+FOB/Prior gastric sleeve.
[23]	String	+FOB/Sub-optimal iron stores.
[24]	String	+FOB/Upper abdominal pain.
[25]	String	+ve FOB
[26]	String	+ve FOBT
[27]	String	+ve HP serology
[28]	String	+ve anti-gliadin Ab, -ve TTG Ab, +ve ANA, iron deficient.Mild

The result can be saved into various formats with a right click on the result element, in this case 'b':



Name	Class	Data
Context[113351]	Sequence(KeyValue)	Sequence{<KeyValue: TKeyValue>,
metadata	Tuple	<Tuple>
keys[113351]	Sequence(KeyType_Test...	Sequence{<KeyType_TestSandbox:
ids[96564]	Sequence(KeyType_Test...	Sequence{<KeyType_TestSandbox:
indications_raw[112462]	Sequence(KeyType_Test...	Sequence{<KeyType_TestSandbox:
$SCOPE1.result[19073]	Sequence(String)	Sequence{$ years mid-abdominal dis
indications_cleaned[19073]	Sequence(String)	Sequence{$ years mid-abdominal dis
Result	Tuple	<Tuple>
a[96564]	Sequence(String)	Sequence{1000004^^^Medical-Obj
b[190		nce{$ years mid-abdominal dis
[1		s mid-abdominal discomfort an
[2		stools for 6 weeks. P/H of GO
[3		s' like sensation. Post prandial
[4		s' like sensation. Sinusitis. Oc
[5		ogical' Barretts. GORD under c
[6		reflux, regurgitation
[7]	String	'choking' sensation related to the up
[8]	String	+Coeliac serology.

A clinical use case is for audit and research. The auditable and researchable Map cache could be updated on a regular basis to surface population clinical data of interest for bespoke Reduce gello queries.

In a gastroenterology practice doing colonoscopies, this includes the 'To caecum' rate (an assessment of the quality of the scope procedure) and an Adenoma rate (the number of adenomas on subsequent biopsy for biopsies performed with endoscopy). The example above is for upper endoscopies. Different clinical queries would be desired.

BACK